New York Computer Science and Digital Fluency: Kindergarten Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **New York Computer Science and Digital Fluency: Kindergarten Course** introduces students to foundational programming concepts through a block-based programming language. Students will develop computational thinking and problem-solving skills while learning to create interactive projects, animations, and games. This course emphasizes creativity and collaboration, providing students with a solid base in computer science concepts and digital literacy.

Learning Environment: This course is teacher-led and includes ready-to-use lessons following a consistent structure: **Introduction, Guided Practice, Independent Practice, Extension, and Reflection**. Instruction follows an "I do, we do, you do" model and incorporates spiral review to reinforce concepts and build confidence over time.

The course includes **36 lessons**, each approximately **45 minutes** long, providing a full year of instruction when taught once per week. While the course allows for instructional flexibility, some lessons are required to fully meet state computer science standards and are clearly identified within the syllabus. Required lessons are labeled with the specific standards they address to support planning and compliance.

Programming Environment: Students will write and run programs that are saved in students' accounts. The environment supports interactive, hands-on programming, enabling students to create and debug projects in a user-friendly interface.

Prerequisites: There are no prerequisites for this course. It is designed to support all learners, regardless of prior computer science experience.

More Information: Browse the content of this course at https://codehs.com/course/NY_K/overview

A clickable PDF can be found at https://codehs.com/NY-K-5Roadmaps

Course Breakdown

Unplugged Exploration

This module introduces foundational computer science concepts through hands-on, screen-free activities. Students explore sequences, events, and variables through physical movement, card games, and collaborative problem-solving.

Objectives / Topics Covered	● Sequencing step-by-step instructions ● Understanding how events trigger actions ● Exploring variables as data containers ● Decomposing tasks into smaller steps
Lessons	**Sequences (Unplugged) (K-1.CT.4, K-1.CT.5, K-1.CT.6)** ● Create a sequence of step-by-step dance instructions to explore how programs follow ordered commands. **Coding Card Game: Sequences (K-1.CT.6, K-1.CT.10)** ● Work together to sequence instructions that move a character through a maze. **Coding Card Game: Sequences 2 (K-1.CT.6, K-1.CT.10)** ● Practice building and refining movement sequences to solve a new maze challenge. **Acting with Events (K-1.CT.6)** ● Act out how an event triggers a corresponding action in a program. **What Is a Variable? (K-1.CT.7)** ● Model how programs store and manipulate data using variables.

Unit 1: Getting Started (5 lessons)

Students build the foundational technology and computational thinking skills needed for the course. They learn to navigate the coding platform, practice mouse and keyboard use, explore how computers work, and begin thinking algorithmically through real-world routines.

Objectives / Topics Covered	● Navigating the CodeHop platform ● Developing mouse and keyboard skills ● Identifying computer hardware, software, input, and output ● Applying computational thinking to real-world tasks
Lessons	**Welcome to CodeHop!** ● Log in and explore the CodeHopJr Playground before starting a full lesson. **Mouse Practice** ● Demonstrate dragging and clicking skills through a series of mouse games. **Keyboard Introduction (K-1.DL.1, K-1.NSD.2)** ● Use letters, numbers, and basic keyboard functions effectively. **Computer Basics: Introduction (K-1.IC.6, K-1.IC.7, K-1.NSD.1, K-1.NSD.2, K-1.NSD.3)** ● Identify what a computer is, how it is used, and how to distinguish input, output, hardware, and software. **Introduction to Responsible Technology Use (K-1.CY.5, K-1.DL.7, K-1.IC.2)** ● Identify ways to use technology safely and responsibly, including understanding an Acceptable Use Policy. **Computational Thinking: Morning Routines (K-1.CT.1)** ● Apply computational thinking to break down, sequence, and simplify steps in a morning routine.

Unit 2: Programming Exploration (6 lessons)

Students dive into the CodeHopJr programming environment through a story-driven series of lessons following Scout the squirrel. They learn to add characters, build sequences of motion blocks, and create animated scenes using events and backgrounds.

Objectives	● Exploring the programming interface

/ Topics Covered	● Adding and modifying characters and backgrounds ● Building sequences using motion blocks ● Using events to trigger actions in a scene
Lessons	**Scout Adventures 1: Introducing Scout** ● Explore the programming interface and add characters to a scene. **Scout Adventures 2: Scout Starts Exploring** ● Add backgrounds and a new page to a program. **Scout Adventures 3: Scout Meets a Friend** ● Delete and modify characters within a program. **Scout Adventures 4: Scout Explores the Forest (K-1.CT.4)** ● Use motion blocks to move characters around the stage in a program. **Scout Adventures 5: Scout and Bluebird Help (K-1.CT.4)** ● Build a sequence of motion blocks to move characters across the stage to collect objects. **Scout Adventures 6: Scout Celebrates with Friends (K-1.CT.4)** ● Create a celebration scene using characters, pages, backgrounds, and sequences with events.

Unit 3: Sequences and Events (7 lessons)

Students apply sequencing and event concepts in the CodeHopJr environment by creating drawings, animations, and interactive programs. They explore a variety of blocks including show/hide, grow/shrink, and data tools to build more dynamic projects.

Objectives / Topics Covered	● Using events to trigger character actions ● Building and combining sequences ● Collecting and organizing data ● Using show/hide and grow/shrink blocks creatively
Lessons	**Drawing Tools: Fairy Tale Painting (K-1.DL.4)** ● Use painting tools to design and create a fairy-tale scene. **Introduction to Events (K-1.CT.6)** ● Create a program that responds to multiple types of event blocks. **Sequences: Snowball Fight (K-1.CT.6, K-1.CT.9, K-1.DL.4)** ● Build a program using multiple sequences to animate a snowball fight scene. **What Can Data Tell Us? (K-1.CT.2, K-1.CT.3)** ● Collect, organize, and analyze data about how students travel to school. **Introduction to Show and Hide Blocks (2 Classes | K-1.CT.6, K-1.DL.4)** ● Use show and hide blocks in a sequence to make characters appear and disappear. **Introduction to Grow and Shrink Blocks** ● Program characters to change size using grow and shrink blocks.

Unit 4: Pages (2 lessons)

Students learn to create multi-page programs, expanding what their projects can do by linking pages together. They apply navigation blocks to build interactive experiences that move across multiple scenes.

Objectives / Topics Covered	● Creating programs with multiple pages ● Using go-to-page blocks for navigation ● Combining events and pages in a program
Lessons	**Introduction to Pages (K-1.DL.4)** ● Create a program that includes multiple pages. **Using the Go To Page Block (K-1.CT.4, K-1.DL.4)** ● Program a go-to-page block to switch between pages in an interactive activity.

Unit 5: Block Exploration (5 lessons)

Students expand their programming toolkit by exploring speed, sound, and personal expression blocks. They also practice debugging, an essential skill for identifying and fixing errors in their programs.

Objectives / Topics Covered	● Animating characters using speed blocks ● Adding sound and speech to programs ● Creating personal expression projects ● Identifying and correcting bugs in sequences
Lessons	**Introduction to Speed Blocks (K-1.CT.4, K-1.CT.6, K-1.DL.4)** ● Create a program that uses different speed blocks to animate characters. **Animal Sounds (K-1.DL.4)** ● Use say or sound blocks to program a character to speak or make noise. **All About Me! (2 Classes | K-1.CT.4, K-1.CT.6, K-1.DL.4)** ● Create a program that expresses personal favorites and interests. **Debugging: Events and Motion (K-1.CT.4, K-1.CT.9)** ● Find and correct bugs in event-based and motion sequences.

Unit 6: Loops (3 lessons)

Students are introduced to loops as a way to repeat code efficiently. They apply repeat and forever loops to create games and animations, discovering how loops simplify programs and enable new creative possibilities.

Objectives / Topics Covered	● Understanding how loops repeat sequences of code ● Using repeat loops in interactive programs ● Applying forever loops to create continuous animations
Lessons	**Loops (K-1.CT.1, K-1.CT.4, K-1.CT.6, K-1.CT.8, K-1.CT.9)** ● Create a program using loops and explain how loops are used to repeat code. **Loops: Frog and Rabbit (K-1.CT.8)** ● Use loops to repeat code and animate characters in a program. **Forever Loops: Fireworks (K-1.CT.8)** ● Create sequences that repeat continuously while the program runs.

Unit 7: Culmination Project (2 lessons)

Students apply all programming skills learned throughout the year to create an original wildlife scene. This capstone project challenges students to combine events, sequences, and loops in a single creative program.

Objectives / Topics Covered	● Combining events, sequences, and loops in one project ● Applying the design process to a creative programming challenge
Lessons	**Wildlife Scene Project (2 Classes | K-1.CT.4, K-1.CT.6, K-1.CT.9, K-1.DL.4)** ● Create an animated wildlife scene that demonstrates mastery of events, sequences, and loops.

Unit 8: Digital Literacy (6 lessons)

Students explore essential digital citizenship and technology literacy concepts, including online safety, the impact of computing, research skills, data storage, and how networks connect people and devices.

Objectives / Topics Covered	● Identifying private and personal information ● Explaining how technology affects daily life ● Using research sources to find and share information ● Understanding how data is stored and networks connect devices

Lessons	**Keeping Information Safe (K-1.CY.1, K-1.CY.2, K-1.IC.4)** ● Identify personal and private information and explain why it should be kept safe. **Impacts of Computing: Daily Life (K-1.IC.1, K-1.IC.3, K-1.IC.7)** ● Describe how computing devices shape daily life and how technology has changed the way people live and work. **Introduction to Research (2 classes | K-1.CT.3, K-1.DL.3, K-1.DL.4, K-1.DL.7)** ● Find information using research sources and create a program to communicate findings visually. **Introduction to Data Storage and Files (K-1.NSD.5)** ● Recognize that computers store data as files and model how data is collected and stored. **Using Networks to Connect (K-1.NSD.4)** ● Describe how people and devices connect and share information using wired and wireless networks.

New York Computer Science and Digital Fluency: Kindergarten Course Supplemental Materials

Resources	**Description**
Parent Welcome Letter (Spanish)	Send this letter home to introduce families to their new computer science curriculum.
Warm-Up Activities	This warm-up activity slide deck provides 5-10 minute problems aligned with computer science skills to engage students at the start of class, allowing teachers to preview or review concepts with answer keys and discussion tips included in the Speaker Notes.
Program Self-Assessment (Spanish)	This is a student self-assessment tool designed to help K-6 learners reflect on their programming projects, evaluate their skills in algorithms, debugging, collaboration, and reflection, and set goals for improvement.
Peer Review Resources (Spanish)	This provides structured worksheets to facilitate student feedback during collaborative coding projects. It encourages reflection by guiding students to highlight successes, ask questions, and offer constructive feedback on their partner's work.
Lesson Reflection & Computational Thinking (Spanish)	This guides students in engaging with computational thinking concepts, preparing for discussions, reflecting on lessons, and applying their learning to real-world problem-solving.
These resources and more are found on the **CodeHop Resources Page**.	